REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
May 29, 2003

3. REPORT TYPE AND DATES COVERED
Final 07/28/99 – 12/31/02

4. TITLE AND SUBTITLE
TRIAD: Translating Relaying Internetwork Architecture Integrating Active Directories

5. FUNDING NUMBERS
MDA972-99-C-0024-P00004

6. AUTHOR(S)
Professor David R. Cheriton

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science Department
Stanford University
Stanford, CA 94305-9040

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency (DARPA)
Contracts Management Office (CMO)
3701 North Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
Office of Naval Research
Seattle Regional Office
1107 NE 45th Street, Suite 350
Seattle, WA 98105-4631

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The TRIAD project addresses a crisis in the Internet architecture that has arisen because the original architecture has been significantly compromised to support content distribution and NAT, reducing the robustness and the security of Internet applications, yet IPv6 does not address these problems fully and shows no immediate prospect of being significantly deployed. This research developed and demonstrated the benefits of a new/revised Internet architecture in which all endpoints and (multicast) channels are identified by name rather than address. This simple change in Internet design leads to a consequential set of changes, including name-based checksums, integrated naming and routing, unified transport, secure internet access, feedback-based routing and a number of other innovations that dramatically improve the availability and security of the Internet without imposing the gratuitous cost of changing to IPv6, which fails to address these issues. TRIAD's revolutionary approach to Internet architecture thus allows multi-dimensional scaling that is particularly suitable for future military needs, including very small-scale embedded systems and distributed command and control.

14. SUBJECT TERMS

15. NUMBER OF PAGES
26

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION ON THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

TRIAD: Translating Relaying Internetwork Architecture integrating Active Directories: Final Report

Prof. David R. Cheriton
PRINCIPAL INVESTIGATOR
cheriton@dsg.Stanford.EDU
(415)-723-1131

May 29, 2003

1 Introduction

This is the final report for ARPA Order No. H719, Program Code No. 9E20, issued by DARPA/CMD under Contract MDA972-99-C-0024. The final approval and distribution of funds occurred at the end of July, 1999 so actual funded work started on August 1, 1999.

TRIAD addresses a crisis in the Internet architecture that has arisen because the original architecture has been significantly compromised to support content distribution and NAT, reducing the robustness and the security of Internet applications, yet IPv6 does not address these problems fully and shows no immediate prospect of being significantly deployed. A key objective of TRIAD is demonstrating the benefits of a new/revised Internet architecture in which all endpoints and (multicast) channels are identified by name rather than address. One specific objective is providing scalable content routing, caching and content transformation, with an explicit *content layer*. Another is to solve the current addressing limitations of IPv4, including limited number of addresses, conflicts with network address translation (NAT), and difficulty in controlling source spoofing, all without having to make a difficult transition, like that to IPv6. Another objective to provide secure scalable solutions to mobility, virtual private networks, and policy routing. We hope to show that TRIAD's revolutionary approach to Internet architecture allows multi-dimensional scaling that is particularly suitable for future military needs, including very small-scale embedded systems and distributed command and control.

The following sections describe the results for each of the original deliverables, plus sections corresponding to what we actually developed, if that was different from what was originally proposed.

2 TRIAD Model

The TRIAD appproach is three-pronged, anchored in the objective of avoiding gratuitous changes relative to the existing Internet protocols, for deployability.

Name-based Identification: TRIAD uses hierarchical character-string names for content endpoint identification, using DNS names and URLs for backwards compatibility. In particular, the endpoints in a transport level connection are identified by name, not address. Similarly, for the endpoints in a secure connection, comparable to IPsec. The names of endpoints are included in the end-to-end transport checksum, not the packet addresses. TRIAD further extends this naming to named multicast channels, as provided in the EXPRESS or single-source multicast (SSM) model.

In this approach, a mobile military platform, for example, establishes end-to-end secure channels based on named identification, allowing these connections to be persistent through transparent recovery at the transport level, as this platform moves between different environments and as network resources fail, are compromised, or are reassigned.

Integrated Naming, Routing and Connection Setup: The "directory" maintained by a router in TRIAD logically maps from content name to next-hop, rather than from address to next-hop. In TRIAD, a packet address is just a routing tag that allows for more efficient forwarding. A TRIAD router advertises or "pushes" name mapping information out towards clients using the same "push" or advertise mechanisms used to disseminate routing information. This contrasts with the "pull" or request-driven approach of conventional DNS. A name lookup is relayed from the requestor to the server, with connection setup information piggybacked on the request, eliminating a roundtrip time for a separate connection setup.

Continuing our earlier example of the mobile platform, this "push" approach to naming allows the platform to locate by name a nearest-by instance of some useful asset, whether informational or physical, with minimal communication. The push of the naming information makes it available locally while the coupling to the routing mechanism allows it to be provided with an indication of proximity. Integration with connection setup minimizes communication to get to the content.

Path-extended Addressing: TRIAD goes beyond the single address space, as was originally envisioned with IPv4, and which IPv6 attempts to return to. Instead, TRIAD accepts the direction that the ad hoc deployment of NAT has produced, of dividing the Internet into a set of address *realms* with address translation between then. For scalable addressing across these realms, TRIAD uses a shim protocol called WRAP that provides a variant of loose source routing on top of IPv4.

As illustrative of its benefits, WRAP allows a mobile military platform, as in the earlier example, to direct its communication through a security-policy specification path while concealing its internal network and addressing structure.

Overall, TRIAD offers an approach to scaling the Internet that addresses key future military communications requirements plus supports important commercial demands, such as content distribution, NAT, and mobile wireless access, yet is sufficiently compatible with, and deployable within, the current Internet to hold strong potential of acceptance.

The following areas describe our key accomplishments, with indication of how they correspond to items in our statement of work for the contract.

3 Name-based Checksum

As another illustration of the system phenomenon that the simplest idea is often the most powerful[1], a key accomplishment in TRIAD is the development of the concept of a *name-based checksum* and the associated proposal [2] to provide this as an incrementally deployable extension to TCP.

The basic idea is to compute the transport-level checksum using the names of the endpoints as a check on correct demultiplexing of the packet, rather than using the standard TCP-defined "pseudo-header" which is computed using the IP layer source and destination addresses. This latter approach, obviously dating from the origins of the Internet, is an unfortunate layer violation that conflicts with achieving end-to-end reliability.

Prior to TRIAD, the conventional wisdom was that NAT interfered with achieving end-to-end reliability and the solution was to eliminate NAT. Of course, any realistic review of Internet directions clearly indicates that NAT is not going away, and in fact is proliferating at an enormous rate. However, in the TRIAD work, we identified that, because of DHCP (as well as NAT), addresses in fact have no real meaning — an address can change which host interface it is bound to over time, including during the time from when a DNS lookup returned the address until the requesting host is done with using this address. Thus, not only does the conventional pseudo-header interfere with NAT, but it does not actually ensure end-to-end reliability. The name-based checksum solves both problems.

[1] Considering the idea of IP as effectively an universal datagram as another.

Our work also demonstrates how this approach can be implemented with the same performance as current TCP checksum computation. The draft RFC also proposes how to implement this as a TCP option so that a TCP connection can negotiate from effectively no end-to-end reliability in the presence of NAT to a stronger than normal form of end-to-end reliability based on the naming. In effect, it adds a key negotiation protocol to TCP that negotiates between the endpoints as well as the intermediate NAT boxes on how to treat the checksum.

The name-based checksum can be viewed as providing a significant step toward unification between message authentication codes (MAC) and packet checksums, viewing the names of the endpoints as a key or input to generate a key for the MAC. It also unifies in the sense that it calls for demultiplexing to the "connection" or socket before performing the checksum computation, the same as for the MAC in a secure protocol.

We are hoping to have our proposal considered by the IETF for adoption as an extension to TCP. Interestingly, it also solves a problem for IPv6. Here, one would like to allow TCP between IPv4 and IPv6 hosts, which effectively requires NAT as well. NBCS supports end-to-end reliability in this setting as well.

As a related part of this work, a Ph.D. thesis [4] and publication [3] was produced reporting on studies that examined the sources of data corruption in a network as well as the frequency with which this corruption might escape detection by various checksum algorithms. These results have been significant in providing the first basis for understanding how important current checksum approaches are, their strengths and weaknesses, and guidance for the design of future protocols.

One issue that arises with the name-based checksum approach is that all communication must be connection-oriented (e.g. TCP) in order to use this end-to-end approach. However, conventional wisdom holds that TCP cannot be used for real-time, and it is clearly not usable as specified for multicast. This realization prompted an investigation of the feasibility of adapting TCP to support these other applications.

4 TCP Extensions

In our work, we took the novel approach of investigating the amount of change required to TCP to support all other applications and comparing that to the complexity of introducing new transport-layer protocols to serve these applications.

In this work, we identified three extensions to TCP:

1. **TCP-RTM:** real-time mode - do not block too long on dropped packets
2. **TCP-SMO:** single-source multicast optimization - support multicast as an optimization for point-to-multipoint delivery.
3. **TCP-framing:** efficient framing support so applications can control how their data units map to packets, including not having the APDUs straddle packet boundaries.

This extended version of TCP has been implemented in the Linux kernel and is being made freely available to other researchers. We have also generated several reports [5, 7] on this work as well as a Ph.D. thesis [8].

With the TCP-RTM extension, we showed that TCP provides recovery of packet loss with fast retransmit without introducing excessive delay in networks with up to 4 percent packet drop, assuming a reasonable round-trip time. We further recognized that packet drop for compressed real-time multimedia data is particularly disruptive to the end-user experience and showed that increased playback delay and fast retransmit provided a superior user experience compared to conventional real-time datagram delivery where there is no recovery from packet drop. Indeed, it is a far better trade-off to use a media compression scheme, such as MPEG-1/2 for video, and packet retransmission (to deal with the higher sensitivity to packet loss) than to send raw video with its much higher bandwidth requirements.

Using this extension, we demonstrated a "phone server" [6] that implemented real-time audio communication using application-level matching of TCP sessions.

With the TCP-SMO implementation, we showed that it was feasible to extend TCP to support single-source multicast and for it to support thousands of receivers directly, limited primarily by ack processing overhead. (A proposed further extension here is to allow lower rates of ack'ing than in conventional TCP.) Using a hierarchical collection of servers, TCP-SMO can support essentially an unlimited number of receivers.

With a TCP extension to support framing, we showed significant improvement in performance for "frame"-oriented communication, including RPC communication.

In sum, this portion of the effort extended TCP to allow all applications and services to operate over (extended) TCP, eliminating the need for so-called "connectless" transport protocols in the Internet. Reliance on having a connection-oriented transport protocol is important for the endpoints to know the names of the endpoints for the NBCS approach and to support our source routing and name-based routing approach, as described next.

5 WRAP

To allow extensibility of addressing without disrupting the established IPv4 routing layer, we developed the Wide-Area Relay Addressing Protocol (WRAP) [9]. This protocol provides more effective addresses than IPv6 yet does not require changes in the IPv4 infrastructure except at *relay points*, which correspond to gateways between realms and optionally between autonomous systems such as separate ISPs.

In short, WRAP is a shim protocol positioned between the IP layer and the transport layer that provides loose source routing as well as route recording. In this sense, it bears similarity to the IP LSR option but differs in not having to be inspected by intermediate routers, a significant overhead on the forwarding path. It also differs in ensuring that the IP source address is always the last relaying node, not the original source, avoiding a security problem with LSR.

The use of NBCS discussed earlier allows WRAP to be used to modify the IP header along the path designated by WRAP while retaining end-to-end reliability. The use of connection-oriented transport based on the extended TCP described above provides a means to recover from a failing source route, whether because of failing relay points, routers, or links. In particular, the connection-oriented protocol can rebind to the endpoint, potentially getting a different source route, without losing the connection.

WRAP was implemented as a module in the Linux kernel, implementing the functionality of a WRAP relay connecting separate address realms. Each connected realm has its own routing table. It was designed to meet the three following objectives:

1. Emulate a hardware router, in order to demonstrate the feasibility of actually implementing WRAP in hardware.
2. Make it easy to add more functionality.
3. Make it easy to use, read, and understand the code, in order to facilitate distribution of the code to other researchers.

The key characteristic, which allows the WRAP module to meet all three objectives, is its simple, three step per packet operation:

1. Classification
2. Processing
3. Transmission

An incoming packet is first sent to a routing table for *classification* The input to the classification process is the relevant portion of the IP and WRAP header. The result of the classification is the outgoing network interface and the type(s) of processing required, i.e. what to do to the packet and where to forward it.

More specifically, the input to the classification process is 72 bits of IP and/or WRAP header content. For pure IP packets, these bits correspond to

1. an 8-bit tag,
2. the 32-bit IP source address, and
3. the 32-bit IP destination address.

For WRAP packets, the 72 bits of input correspond to

1. 8 bits from the IP destination address, used as a tag,
2. the 32-bit IP source address, and
3. the next 32-bit IRT component.

Thus, packet classification involves a single lookup based on one 8-bit tag and two 32-bit addresses.

There are three types of packet processing and each type is implemented as a separate processing entity:

1. The *filtering entity* sends incoming packets to a filtering table. If a match is found, the packet is dropped.
2. The *tunnelling entity* handles packets that need VPN encryption or decryption.
3. The *relaying entity* handles all WRAP packets. The only processing it does is to swap the IP source and IP destination addresses with the appropriate reverse and forward IRT components respectively and update the WRAP pointer specifying the beginning of the forward IRT. Thus, the relaying module touches only four words of the IP/WRAP header plus the WRAP pointer.

The output of the classification process includes three flags. Each flag corresponds to a processing entity. If the first flag is on, the packet should go through the filtering entity, if the second flag is on, it should go through the tunnelling entity, etc. It is possible that two or all three flags are on at the same time, i.e. a packet has to go through two or all three processing entities – but always in the above mentioned order. This linear structure mimics state-of-the-art hardware design patterns, and helps justify our assertion that WRAP is feasible to implement in hardware.

The output of the classification process also includes the outgoing network interface. This may be a physical interface or a *virtual* interface, i.e. a separate module where the packet being treated should be forwarded. The capability of specifying a virtual interface as the outgoing network interface is used to forward packets to the WRAPID compatibility module (described in Section 7).

Our WRAP implementation was designed to emulate hardware router operation. Specifically:

1. The classification process maps directly to an efficient hardware implementation using a ternary CAM with a 72-bit wide label field.
2. Each processing entity could map to a hardware module. Then, the three flags of the classification output would signify which of the hardware modules should be enabled.
3. Finally, virtual interfaces could map to linecard addresses. E.g. the WRAPID module could be implemented as a separate linecard.

Our implementation leads to a single lookup per packet in the common case; Common IP and WRAP packets need only go through the lookup of the classification process. A second lookup is required if the packet is coming from an insecure network and has to go through the filtering entity. Also, packets exchanged between WRAP and non-WRAP hosts have to go through the WRAP module routing table twice: once before translation, in order to be forwarded to the WRAPID module, and once after translation, in order to be forwarded to their destination. Although this "double lookup" adds to the translation overhead, it has two important advantages:

1. It decouples WRAPID from WRAP operation, thus adding flexibility; WRAPID can be a linecard in a WRAP relay or a totally independent machine.

2. The alternative would be to do a single lookup, which would involve more than three fields. That would reduce translation overhead, but it would also increase the required label field width of the CAM. With the "double lookup" a 72-bit label field width is enough.

Although we did not implement WRAP in hardware, the software implementation as described above together with our familiarity with hardware router implementation demonstrates the feasibility of implementing WRAP in modern ASIC technology supporting very high speed network links.

While the original focus with WRAP was providing addressing across NAT realms, we further recognized that WRAP provided a significant capability for traceability of packets, prompting an exploration of scalable packet filtering, as described next.

6 WRAP Filtering

As part of this work, we considered Denial of Service (DoS) attacks, a very important Internet problem, which requires an automatic network-based filtering solution. A network layer solution is required because end users have no way to protect their tail circuit from being congested by an attack. Furthermore, filtering has to be automatic because increasingly sophisticated attacks are rapidly making manual response infeasible.

Most routers already offer good filtering capabilities. However, filter propagation is currently manual: The operator on each site determines the necessary filters and adds them to each router configuration.

At a high level, our solution is straightforward: A router is configured with a filter to drop packets and then, if it is dropping a significant number of such packets, it uses a filter propagation protocol to request that the upstream router block this traffic, thereby automatically propagating the filter and stopping the unwanted traffic.

The real problem is how to efficiently manage the limited resources available to a network operator to provide the necessary filtering support: The number of filters per router is bounded by maximum hardware table sizes and is typically limited to several thousands at most. Moreover, there is a processing cost at each router for installing each new filter, removing the old filters, and sending and receiving filter propagation messages.

One key characteristic of the WRAP addressing protocol is that the entire Administrative Domain (AD) path is specified inside the WRAP header of each packet. This leads to a very important fact: Each node in the AD path can look at an incoming packet and determine which networks this packet has gone through. This fact enables the definition of the Active Internet Traffic Filtering (AITF) model [10], which provides effective filtering of attack traffic for the clients of a network with a reasonable provisioning of filtering resources within the network.

6.1 AITF

An AITF network has a *filtering contract* with each network or host it connects to. Such a filtering contract specifies:

1. The ingress filtering request rate R_{ing} at which the network accepts filtering requests to best-efforts block some traffic to the other network/host.

2. The egress filtering request rate R_{egr} at which the network can send filtering requests to get the other network/host to best-efforts block some traffic from coming into it.

A *filtering request* is a request to best-efforts block a flow of packets – all packets matching a specific wildcarded flow label – for the next T time units.

AITF filter propagation occurs at the AD level i.e., filtering requests get propagated from AD to AD. Thus, when one node wants to send a filtering request to another node in a different network, it propagates it through all the nodes in the AD path. For security reasons, it cannot send the request directly to the target node.

On the contrary, filter installation does not happen on every hop of the propagation path. It happens only one hop away from the requestor[2] and the target[3] – in other words the only filtering points are the service provider of the victim and the service provider of the attacker. The service provider of the victim installs a filter only temporarily, to give some time to the service provider of the attacker to respond. The service provider of the attacker is expected to filter its misbehaving customer effectively. If the first attempt to block the attack does not work, i.e. undesired traffic still goes through, filtering becomes more coarse-grained by one AD level: the victim's network disconnects from the attacker's network for T time units. If this fails as well, filtering becomes even more coarse-grained: the victim's local ISP disconnects from the attacker's local ISP for T time units and so on.

Thus, the mechanism proceeds in steps. In every step, it optimistically guesses the last point of trust and tries to push filtering of undesired traffic back to it. If it fails, it pulls the last point of trust further away from the attacker. Eventually, it identifies the true last point of trust and causes undesired traffic to be blocked at that point. Every time the mechanism takes a new step we say that it *escalates* to a new level. Every time this happens, filtering becomes more coarse-grained by one AD level.

The AITF model is an attractive solution to DoS attacks:

1. Attack traffic is blocked within a round-trip time (plus filter installation time) from the requestor to the next filtering point.

2. The requestor can reclaim its filter after this round-trip time, limiting the number of filters it requires to the number of outstanding propagation requests it has.

3. Filter installation happens mainly on the edges of the Internet, where it is feasible to have more filters per attacking host, at least compared to the backbone of the Internet. Only when the edge nodes fail to filter does filter installation "move" towards the backbone.

Based on the limited filtering request rates, a router can limit the CPU cycles used to process filtering requests as well as the number of filters it requires. An analysis of the number of filters required by an AITF node and of the quality of the filtering services provided to its customers as a function of the allowed filtering request rates is straightforward and shows that the AITF model scales to large DoS attacks. E.g. it is straightforward to show that:

1. If an AITF network/end-host is allowed to send 100 filtering requests per second to an upstream provider, then it can face up to $6,000$ simultaneous undesired flows through that provider and reduce their effective bandwidth by a factor of 0.00034.

2. On the other hand, in order to sustain 100 filtering requests per second from each customer network/end-host, an AITF provider needs only 100 total filters per customer.

   These are example numbers. The assumptions that lead to these numbers and the corresponding analysis is provided in a report [10] we generated on this subject.

Finally, the AITF model offers an economic incentive to providers to protect their network from the inside by employing appropriate filtering. Here is why: When provider X is asked to filter a misbehaving client, it can either spend the resources to do so or ignore the request. However, ignoring the request will result in the requesting network being dissatisfied and disconnecting from provider X. Thus, by ignoring filtering requests, provider X limits its connectivity and, thereby, its quality of service. In this way, the quality of a provider's service is now related to its capability to filter its own misbehaving clients, which encourages effective filtering.

In sum, with this work we demonstrated that WRAP could be used to significantly improve the ability to rapidly and automatically filter traffic in the Internet, addressing the growing problem of DDoS attacks.

[2] The *requestor* is the node that originally sends the filtering request i.e., the victim of a DoS attack.

[3] The *target* is the node whose traffic the requestor wants to block i.e., the attacker.

7 WRAPID

We also explored how to deploy WRAP incrementally by developing software that would translate between WRAP-aware hosts and conventional IPv4 hosts, referred to as WRAPID for WRAP to IP Device. WRAPID was implemented as a Linux kernel module, demonstrating the feasibility of unmodified IPv4 hosts to communicate with WRAP-extended hosts and an Internet extended to support the TRIAD extensions in general.

Additional details are provided below.

7.1 Wrapping

Suppose non-WRAP host S wants to communicate with WRAP host T. S does a name lookup on T. The name lookup goes through S's gateway G – which is running both a WRAP and a WRAPID module. G picks an IPv4 address, say I, from a pool of designated "special" addresses and answers S's request. At the same time, G creates a mapping from I to T's full IRT. G's routing tables are configured to route packets addressed to "special" addresses to the wrapping submodule.

Thus, when S sends a packet with I as the IP destination address, G forwards the packet to the wrapping submodule. The mapping from I to T's full IRT is retrieved and used to generate an appropriate WRAP header. Then, the new WRAP packet is sent back to the WRAP module as if it were just received.

7.2 Unwrapping

Now suppose that WRAP host T wants to talk back to non-WRAP host S. T does a name lookup on S. As a result, it receives an IRT leading to S. This IRT ends with G's address, a "special" token and S's IPv4 address. G's routing tables are configured to route packets with "special" tokens as their next IRT component to the unwrapping submodule.

Thus, when T sends a packet with the given IRT, G forwards the packet to the unwrapping submodule. The unwrapping submodule removes the WRAP header, extracts S's IPv4 address from the IRT and uses it as the IP destination address to build an appropriate IP header. Then, the new IP packet is sent back to the WRAP module as if it were just received.

This facility can be deployed as extended NAT boxes in the locations that these NAT boxes typically reside now, namely at realm boundaries (i.e. as firewalls). Thus, WRAP can be incrementally deployed while providing benefits at each stage in terms of improved addressing (into NAT realms) and improved filtering capability, as described in the previous section.

8 Name-based Routing

The use of names as the true identification of Internet endpoints in TRIAD motivated our contract work to develop secure, reliable, and scalable name service. In fact, this requirement is to some degree self-evident even outside of TRIAD because names are the "user-level" identifiers that end-users expect to deal with. If the naming service fails, the Internet is down as far as most users and applications of the Internet are concerned.

The TRIAD "content layer" integrates directory services with the routing infrastructure to address the problems of availablity, security, and reliability of naming. Our solution is called Name-Based Routing (NBR). We briefly summarize NBR below and describe our specific accomplishments under this contract.

A robust network application cannot rely upon addresses alone because addresses have no inherent semantics. Names, by contrast, can have semantics assigned to them; for example, we expect "www.cnn.com" to name content that comes from the Cable News Network. This assignment of semantics to a name is relatively stable and persistent; it is also necessary to isolate applications from changes in network configuration and behavior. But, this layer of indirection between names and addresses requires extra mechanism provided by the network to convert between a name and the set of network resources needed to access the content or service identified by that name. How can we best make this service efficient, scalable, and resistant to denial-of-service attacks?

Traditionally networking has separated content location into two services: naming and routing. Name lookup (performed using DNS in the Internet) converts a name into a lower-layer address. Routing (performed via BGP) converts an address into a sequence of network hops. Name-Based Routing combines these two operations into a single protocol. A key insight in TRIAD is that the process of selecting among multiple servers for content is identical to the process of selecting among multiple paths for a packet. A name-to-address mapping is useless if the network cannot deliver packets destined to that address. Similarly, address reachability is useless if the name an application uses cannot be resolved.

In TRIAD, BGP-level routers are augmented with naming capabilities to become "Content Routers". A content router (CR) participates in a dynamic routing protocol exchanging name reachability information similar to (or even implemented on top of) the BGP peering sessions it already participates in. A content source advertises the names for its content or services to the nearest content router(s); from there the advertisement is propagated to the rest of the Internet.

A name lookup proceeds hop-by-hop along the path of content routers between the client and the server. Each content router consults a "name routing table" built using the content advertisements, chooses the best path toward the content (based on policy and/or measurements), and forwards the request one hop further along that path. The server which ultimately receives the request answers with its own address; this response travels back along the same path, allowing each CR to modify the address. When using WRAP, this means that each CR puts its own address at the start of the path. This solves the problem of generating WRAP addresses in a scalable fashion (i.e., without global knowledge of the Internet topology), and allows routers flexibility and control over the assignment of addresses and paths. The path that the name lookup discovers is the same one used to transfer data, ensuring fate-sharing between name lookup and content transfer. Client have some degree of control over this path, by indicating in their request specific servers or routers they wish to avoid. This allows client applications to avoid malfunctioning or malicious nodes that have not yet been (or cannot be) detected by the network.

The key challenge to scaling NBRP is to reduce the number of advertisements to an acceptable level. The solution is to separate long-term topological location information from short-term routing fluctuations. In NBRP, this is provided by "explicit aggregation". A name may be advertised individually or as part of a named aggregate. A single update arriving for an aggregate name is logically treated as if it were advertisements (with identical paths) for each of the 100s or 1000s of constituent names. Thus, a content router must learn the contents of an aggregate once, but can then use this information for many routing updates. For example, a web hosting facility might aggregate the names of all its customers into a single name advertised to the upstream ISPs. Internal to a content router the routing table entries for an aggregated name are aliases for the aggregate as a whole to the rest of the Internet.

Finally, NBR provides defenses against denial-of-service attacks based on packet flooding. DNS is vulnerable to such attacks because it relies on a small set of root and TLD servers, and allows any client to communicate with any name server. NBR name lookup traffic is constrained to follow the organization of CRs, to it is not possible to directly attack a specific machine. Further, there is no central server providing delegations, so that attacks against remote resources do not affect the availability of local names. To prevent spoofing of advertisements, servers wishing to advertise a name must provide proof (either offline or via a cryptographic operation) that they have a right to that name. Additional techniques such as feedback-based routing (described in Section 9) or end-to-end verification of content are necessary to protect against malicious content routers.

In the work on this contract, besides developing the above design of NBR, we have developed a prototype implementation of NBR, and demonstrated its use in both WRAP and non-WRAP contexts. We have studied the effect of explicit aggregation using real routing table and naming information, and performed simulation of large-scale content routing behavior as reported in one conference paper [11] and a forth-coming Ph.D. thesis [12]. An additional paper [13] explores the security benefits of NBR. We are preparing a revised implementation of NBR which we hope to release to a broader audience.

We have shown name-based routing lookup provides low latency and high availability because it uses only resources on the path to the content. Thus, latency is approximately equal to one RTT to the server, in contrast to DNS or peer-to-peer mechanisms, which may require (on a cache miss) queries to several remote servers. Content distribution networks (CDNs) such as Akamai suffer from a recursion problem in that they rely upon DNS-based mechanisms to direct a client to the best HTTP server. But, the choice of DNS server can be critical for performance and reliability; naming itself should be part of the CDN. Name-based routing, by providing naming as a network-integrated service, avoids both these problems.

We also showed that NBR can be implemented to scale in memory, router processing, and packet traffic to the scale of the Internet, present and future. In particular, NBR is implemented using memory-based hash tables at minimal cost. Our experiments show that name lookup cost is dominated by packet processing, not by the name routing table, which is responsible for less than 10us of latency. The total size of the name-routing table is in the region of a few 100 megabytes to include the total second-level DNS directory, which is affordable in a BGP class router even if the size increased by a factor of 10. Further, explicit aggregation reduces the number of distinct routes to that of BGP or fewer, so that update traffic is minimal. The number of new names (and new servers) being added is comparatively modest, so that changes to aggregates happen at an acceptable rate. Thus, the overall routing traffic is comparable or less than current BGP.

Stepping beyond the original scope of the contract, we recognized Name-Based Routing is more resistant to denial-of-service attack than DNS or peer-to-peer networks. In fact, we made the general observation that any server with an unknown set (and thus unbounded set) of clients *cannot be defended* against a DDoS. Why? An attacker can compromise a large number N of hosts on the Internet and cause them to send valid requests at the limits of the legitimate rate to the service, overwhelming the server for sufficiently large N and making the service effectively unavailable for the true clients. The root DNS servers have precisely this property and thus this exposure. Amassing an "army" of several hundred thousand zombies to attack the root DNS servers is feasible and sufficient to dramatically degrade root DNS service.

In contrast, NBR's topology matches that of the real network so that it is feasible to restrict the direct clients of each NBR router to those nearby in the topology, to build trust relationships (and secure connections) to these specific clients, and then to identify misbehaving clients or routers by monitoring their behavior.

We also showed that NBR's property of caching the entire second-level DNS name database allows it to distribute attack load across content routers so a large number of distributed attackers cannot overwhelm NBR (and a large number of local attackers can only disrupt the local operation.) This is the case because the complete caching means that a randomly generated query (invalid name) produces no load on the broader network; The local NBR router can detect it is invalid and respond immediately and locally. In contrast, a randomly generated DNS query causes a local cache miss which causes a request to be sent the root name server, loading that central and limited resource. Caching these invalid names in infeasible and useless in an attack because the attacker never uses the same name again.

The growing awareness of the importance of attack resistance during the period of the contract and our recognition of the exposures that the current DNS and routing systems have led us to investigate a new approach to routing, as described below.

9 Feedback-based Routing

During our work on this contract, it became evident that, although the Internet has become critical to the economy and national security, the inter-domain routing system, the foundation of Internet communication, is highly vulnerable to various forms of attacks. In particular, a single malicious router can shut down the entire Internet by propagating false routing information. There has been some work using conventional security techniques to attempt to "secure" routing information (for instance, Secure BGP). However, it is unreasonable to assume that no valid router can be compromised. We explored a technique called *Feedback Based Routing (FBR)* to address the robustness problem in inter-domain routing.

Briefly summarizing our separate reports on this work, [14, 15, 16] we first describe the problem in more detail and then our accomplishments in this area.

9.1 The Routing Problem

Inter-domain routing entails coordinating the routers of many administratively independent ISPs to agree on operational routes such that packets are delivered reliably and efficiently. Today, Border Gateway Protocol version 4 (BGPv4) provides this coordination, allowing routers to form peering sessions and exchange reachability information about edge networks. By advertising an AS path to a prefix, a router is claiming it *can* and *will* forward packets to the specified prefix by the advertised next hop. Each BGP router relies on this information to calculate a route to every prefix. However, if the reachability information is incorrect, packet delivery and thus some portion of the Internet can fail.

With the world-wide scale of the Internet, it is infeasible to assume that all ISPs can be fully trusted or that all links and routers remain operational. Links can fail and routers can fail or be compromised by attackers. Both of these events can cause the reachability information provided to a router through BGP to be inaccurate, which in turn means that packets are forwarded incorrectly and not delivered. The more extensively compromised the reachability information is, the larger the portion of the Internet is rendered inoperable.

The vulnerability of the Internet to incorrect reachability information has been made frightfully evident by a number of incidents. For instance, in 1997, a misconfigured BGP router at a small Virginia ISP advertised that it could provide a good route to every prefix in the Internet, causing other routers to divert traffic to this "blackhole". The operation of the Internet was disrupted for two hours. Researchers have shown that smaller scale "blackhole" incidents are occurring on average 15 times per day. Although it is feasible to program inter-domain routers to reject reachability information that is clearly wrong, such as one might expect from accidental misconfiguration, the real challenge is to extend inter-domain routing to defend against incorrect reachability information which is maliciously generated by knowledgeable attackers designed specifically to pass local credibility checks yet still compromise Internet packet delivery. Even just corrupting reachability information so as to significantly increase the delay of packet delivery, either by directing the traffic to suboptimal paths or by orchestrating hot spot congestion, is sufficient to disrupt many applications and the dependent organizations.

With the deployment of very high-speed routers and the availability of long-distance fiber, the Internet is performance-capable of handling a wide range of important applications, including VoIP and enterprise mission-critical VPN service. However, the inter-domain routing vulnerability precludes the use of the Internet for certain critical distributed applications and places organizations that do depend on the Internet at significant risk. Arguably, the lack of attack-resistance in wide-area routing may be the key limiting factor in the utility of the Internet at the present and conversely the greatest risk/concern to Internet-dependent organizations.

In contrast, *Feedback Based Routing (FBR)* can make inter-domain routing resistant to attacks and Byzantine failures. In fact, exploiting the source-routing capability in TRIAD (ideally used at the autonomous system level), we show how critical communication can achieve zero fail-over time with high probability.

9.2 Feedback Based Routing Design

The key idea behind feedback-based routing is for a router to monitor packet traffic on its routes and use this as feedback to determine the usability of the routes. A router accepts reachability information from other routers as *hints*. It does not have to trust this information because it selects which path to use based on how each path actually performs. By treating reachability information as hints and having an additional feedback mechanism, inter-domain routing can be made far more attack resistant than the "shared world view" approach at the core of BGP and most routing protocols.

For simplicity of description, we define an *Autonomous Routing Domain (ARD)* to be a network whose internal packet forwarding and topology is not visible to the outside world, similar to a BGP autonomous system. (We use this separate term to allow us to define it independently of the connotations of *autonomous system*.)

An ARD has business relationship with other ARDs or *edge networks* to forward packets for each other. We call such a business relationship a *link*. The edge networks are represented by prefixes. We call the border routers of an ARD *transit routers*. Each edge network has one *access router* that forwards packet to the upstream providers. The components of the inter-domain routing system are illustrated in Figure 1.

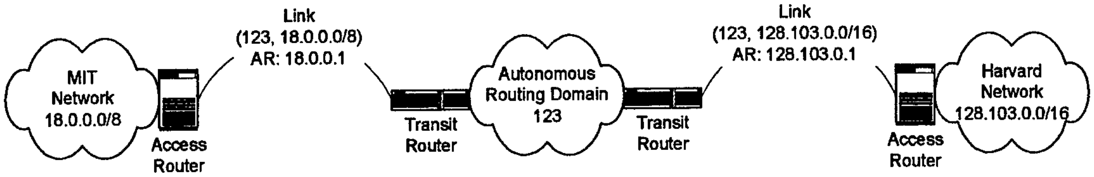

Figure 1: Terminology

An access router learns the Internet topology at the ARD granularity using the ARD Topology Protocol (ATP). ATP is similar to a link state routing protocol, such as OSPF. However, the end points of a link in ATP are autonomous routing domains or edge networks. ATP is designed to propagate only topology information. The operational state of the link is not propagated. This design reduces the number of routing messages significantly. Another consequence of the design is that there is no "withdraw" message in the system. This prevents a malicious router from removing valid topology information.

Each access router specifies the path to a given destination that its packets are to take using "source routing" techniques. That is, it inserts a "routing directive" into the packet. The transit routers along the path look at the routing directive inside a packet and determines how the packet travels inside the ARD in order to reach the next hop ARD. The access router of the destination network removes the routing directive. This source routing is carried out at the level of autonomous routing domains.

Based on the topology information it has, an access router computes two routes to every prefix. The two routes are chosen to be as independent as possible. In the ideal case, they are *ARD disjoint*. There is enough redundancy on the Internet today that two ARD disjoint routes can usually be found. We showed, by processing the Route View data from April 1st, 2002, that there are 4730 multi-homed edge autonomous systems, and there are two AS disjoint routes between 4715 of them.

We define a route to be *unusable* if no packet can flow through it or if its latency, loss rate, or throughput does not meet the standard set by the administrator. An access router monitors the round trip time (RTT), loss rate and throughput on both routes. These metrics are compared with the tolerable range set by the administrator to determine whether a route is usable. When a route becomes unusable, an access router tries to identify the culprit. Once the cause of the problem is identified, an access router computes a route that is disjoint from the usable route and the problematic ARD/link.

If an access router switches to an alternative route once it detects that the current route has become unusable, there is a time window in which packet forwarding stops or suffers significantly. We call this time window the *fail-over time*. The route switching is called *explicit fail-over*. The fail-over time can be reduced to zero using a proactive approach: "simultaneous transmission". In this scheme, an access router duplicates the packets going to certain prefixes and/or using certain protocols, and sends them using different routes. The administrator is responsible for configuring the class of packets that get duplicated. Standard techniques such as described in RFC 2085 can be applied to eliminate the duplicate packets.

9.3 Accomplishments

We implemented the above design in Linux. The first version of the implementation consisted of 16000 lines of C++ code and another 24000 lines of support libraries. A second version of the implementation is under development as part of on-going research.

We further showed that FBR is highly resistant to attack. In particular, there are two types of attacks that can be launched against the routing system. The first type of attack involves the propagation of bogus topology information with the goal of fooling some routers into using a link which either does not exist or does not function well. FBR is resilient to such attacks because an access router does not use a route until it has evaluated its quality, so a non-working or poor performing route is rejected.

The second type of attack involves the propagation of bogus topology information as well as providing misleading feedback to the access routers. However, it takes bandwidth to provide bogus feedback. We showed that the attacker must have bandwidth proportional to the combined bandwidth of the victims to successfully deceive a majority of the access routers. Thus, an attacker has to compromise a significant portion of the Internet to have a significant effect on the Internet routing.

We also performed analysis and evaluation to show that FBR is scalable from the standpoint of resource demand on the control planes of transit and access routers as a function of Internet size.

An access router has to calculate routes to every destination prefix on the Internet and has to monitor the performance of packet forwarding. We showed that the cost to compute the two disjoint routes required by FBR is reasonable [16]. FBR also has the property that if an edge network fails to upgrade an access router, the only hosts that suffer are those whose traffic flows through this access router. The rest of the Internet is not affected. In contrast, BGP requires the cooperation of every BGP router. If some of the routers become overwhelmed with routing updates and fall behind in processing the updates, the entire routing system takes longer to converge. Therefore, no matter how fast an ISP upgrades its BGP routers, the service quality it provides is dependent on the actions of other ISPs. In FBR, if an ISP aggressively upgrades its access routers, its customers directly benefit from the better route selection and faster computation of additional paths.

A transit router is relatively simple. It does not do any route calculation, and only needs to keep track of recently heard links and send new link advertisements to its peers. The cost of doing this is low compared to the computing power of current desktop computers and substantially less than BGP. Moreover, even if for some reason, a transit router falls behind in propagating link advertisement, it is unlikely to affect end-to-end communication.

Another component of scalability is the amount of damage that an attacker can inflict on the Internet relative to the effort expended, as a function of Internet scale. We argue that a routing system is not scalable if the amount of damage an attacker can inflict for a given amount of effort grows more or less proportional to the size of the Internet. This is because attacking becomes increasingly attractive as the Internet grows, and defenders have to expend increasing amounts of effort to protect the Internet. By this definition, the current BGP protocol is not scalable because the effort to compromise one BGP router can pay off in the ability to disrupt almost the entire Internet. In contrast, FBR is scalable because compromising a router only compromises the traffic going through this router and a number of victims proportional to the bandwidth this router has access to.

We have further investigated the deployability of FBR, recognizing that the Internet has to incrementally transition from the BGP model of a shared "world view" of Internet topology to that of individual access routers creating their own view of the Internet from feedback in conjunction with reachability information. Our approach is for ISPs to incrementally enable support for source routing and form peering relationships with other ISPs who have enabled source routing.

FBR has an encouraging synergy with other aspects of TRIAD. In particular, the Wide-Area Relay Addressing Protocol (WRAP) is used to support loose source routing over the WAN at high speeds. Other routers see WRAP packets as normal option-free IPv4 packets. Thus, only the transit routers and access routers have to be upgraded to handle WRAP, in contrast to other alternatives such as the IP LSR option which would cause enormous performance problems on intermediate routers.

In the initial phase of deployment, the legacy portion of the Internet that does not support source routing is treated like an autonomous routing domain with ARD number 0. Every FBR enabled ISP peers with it in the initial phase. Each ISP who joins the FBR camp is assigned an ARD number that is actually a "virtual" IP address. This address is reachable from the legacy Internet. An access router can learn the existence of source routing enabled ISPs through a variety of methods, such as routing registries or DNS.

The benefit of initial deployment is the decrease in fail-over time. One of our future research directions is to quantify this benefit. Such a deployment scenario is similar to what some current route control systems, such as products from Route Science, do for certain customers.

When more and more ISPs start supporting source routing, the portion of the Internet that does not support source routing will have shrunk significantly. Some routes will be free from the legacy Internet. These routes will not be affected by the long BGP convergence time, thus motivating ISPs to stop peering with the portion that does not support source routing.

9.4 FBR Summary

We have shown that FBR is scalable, and in particular more scalable than BGP because:

- The damage an attacker can inflict is bounded in proportion to the number of routers and links it can compromise, whereas in BGP it is not.
- The service level provided by an FBR router to its directly attached customers is primarily determined by its own capacity and connectivity and largely independent of the scale of the Internet, whereas BGP convergence makes the performance dependent on the routing capacity of *all* routers.
- The resources required by an FBR router remain effectively constant as the Internet grows, given the historical improvement trends in processor and memory costs.

In particular, we show that FBR rejects incorrect reachability information that would have otherwise caused packet traffic to be mis-directed in a conventional routing system based on BGP or any similar "shared world view" system. We further show that an attacker requires bandwidth proportional to the combined bandwidth of the victims to generate sufficiently misleading feedback to disrupt packet traffic, so the level of disruption is limited by the number of routers that an attacker can compromise.

Limiting routing decision making to individual access routers means that the investment an ISP makes in such an access router is justified by being directly beneficial to its customers. An access router is not dependent on the speed with which the routers of other ISPs converge in their routing calculations. We note that FBR achieves scalability by giving control of route selection to the edge networks and freeing the core routers from maintaining a complete model of the Internet. This is in the spirit of the Internet end-to-end principle, allowing the ends to implement the desired service.

While FBR was not fully anticipated in the original proposal, it has tremendous synergy with the other aspects of TRIAD; we are planning to pursue further aspects of this direction as part of on-going Ph.D. thesis research.

10 Secure Internet Access

As part of the TRIAD work, we examined the highly touted supposed "compelling reasons" for IPv6 deployment as indications of areas that we needed to consider as well in developing TRIAD as a next generation Internet architecture. One such area is mobility. According to IPv6 proponents, IPv6 is needed for the very large number of mobile IP devices expected in the future. However, this claim is based on the expectation that each such device would have its own public (world-wide unique) IP address. In contrast, current cell phone technology, for instance, dyamically assigns a new address to a device when it enters a realm under the control of a new cell controller.

In our investigation of this area from a TRIAD perspective, we concluded that it was feasible to use a similar approach, namely dynamically assigning a private NAT'ed IP address to a mobile device when it enters a wireless realm. In fact, it is arguably superior to do this because then there is no need to update the routing infrastructure to handle a "foreign" address or to support an IP-level redirect (with its associated security concerns) as some have envisioned in the IPv6 mobile environment. Because the real identification of endpoint is at the name level in TRIAD, addresses act only as transient routing tags, so assignment of new addresses has no real effect on host identification. Moreover, the cost of assigning an address is insignificant compared to the cost of properly authenticating a host that has just arrived in a new wireless realm. This realization motivated an investigation of how to provide secure wireless access to networks. We further realized that the problem is not unique to wireless; For instance, a company may have wired access ports in conference rooms which visitors can access. For security reasons, access through these ports should require authentication as well.

Section 10.1 describes our security architecture, put in the perspective of mobile computing. Our architecture solves the vulnerabilities related to WEP and avoids major drawbacks related to standardized protocols such as IEEE 802.1X, DHCP, and EAP.

Section 10.2 describes an ongoing effort to provide a self-calibrating user tracking system that uses lateration based on signal strength readings in a IEEE 802.11 network. The major contribution of this work is the design of a system that minimizes its reliance on humans during the calibration phase[4].

Section 10.3 describes the motivation for using a system like the one described in section 10.2 to geographically limit the range of a wireless network. Such a system provides administrators with the power of defining the exact geographic areas to be used by mobile clients to authenticate themselves and consequently access the protected local network. This project addresses the problem of war-driving attacks, so well covered by the media today, and also helps detect rogue access points, another known risk of deploying 802.11 networks.

10.1 Secure Mobile Computing

This investigation addressed the problem of providing secure network access for mobile users while having two seemingly contradictory objectives in mind: to maximize security and to minimize the overall system complexity and number of protocols involved. Taking wireless networks as an example, mobile users are faced today with two main kinds of network settings: totally open networks, such as the ones deployed in cafeterias and other public places, and the "secure" installations, which seem to increase the number of protocols involved every time a vulnerability is found. It is not hard to see that neither of these solutions satisfy both clients and network administrators.

From the user's point of view, the access should not only be secure, but also convenient, so there is no need to manually configure the laptop or enter special codes or create a new account at each new location. On the other hand, network operators need to be able to provide this secure Internet access without compromising their network security or introducing excessive management overhead. For instance, an employee hosting a visitor at a company location should be able to provide Internet connectivity without giving the visitor access to the internal company network and without performing too many configuration steps.

We consider access control to be a distinct problem from end-to-end (E2E) security. While the service provider is completely oblivious to how E2E security measures work, it is highly interested in controlling who gets to use its network infrastructure. A faculty member from Berkeley visiting Stanford will never be able to set up a secure E2E connection with her mail server in Berkeley if she has not satisfied Stanford's requirements, a necessary step to get network connectivity.

The problem is then to provide network administrators with secure and effective means of performing fine-grained access control while providing mobile clients with an easy-to-use, interoperable solution. We expect a good solution to provide a superset of the following properties: mutual authentication, flexible authorization, access verification, interoperability, simple user interface, and data confidentiality and integrity. We show that this can be achieved with a limited number of protocols and services.

As work on this contract, we designed and implemented an architecture that provides mobile users with secure network access while allowing network administrators to perform fine-grained access control [17]. Our architecture was designed with two main objectives in mind. The first is to provide additional services, such as transparent connectivity migration between different networks, client differentiation, and ease of use. The second, and most

[4] The calibration phase occurs after the access points have been placed in the environment but before clients can actually be located by the system. The accuracy of existing systems heavily depends on the calibration phase.

important, is to provide all these services with the smallest number of algorithms and protocols. When dealing with communication protocols, especially when dealing with network security, less is better: more compact systems are easier to design, analyze, and deploy. The advent of DoS attacks just makes this point clearer: every service added may be used to implement DoS attacks.

In our solution, a mobile client is provided with two main services: a secure way to establish session keys, implemented by a Diffie-Hellman-based authenticated key establishment algorithm, and a secure tunneling service that uses the fresh session keys to provide privacy, integrity, replay detection, and sender authentication over link-layer frames. Our design philosophy is simple and provides strong security: a mobile client has to perform an authentication handshake with each access point it wants to use, establishing a different set of session keys with each one. These fresh session keys are then used to provide privacy, integrity, and replay detection over the frames transmitted over the wireless link. In this system, there is no need for "back-end" authentication servers, as all necessary services are securely integrated by the protocols run at the access points.

SIAP, the *Secure Internet Access Protocol*, provides a Diffie-Hellman-based authenticated key establishment service, using RSA keys as means of authenticating users and access points. Each client and access point has a set of tuples comprised of a domain name *dn* and a key pair *kp* (e.g. RSA), used for authentication purposes[5]. By the end of a SIAP handshake (maximum of 4 messages), a client is provided a secure state that includes the fresh session keys, its IP address, and other authenticated configuration data, such as the IP addresses of the default gateway and DNS servers. Together with its IP address, a SIAP client receives a lease, defining a period of time during which the client does not need to perform another SIAP handshake.

SIAP entities authenticate each other by the use of RSA key pairs. Our current implementation uses a per-domain certification authority that binds a client's (or AP's) domain name to a key pair by means of a digital signature. We are studying other alternatives, such as the use of online authentication servers, in order to reduce the dependence of the architecture upon certificates and to provide effective revocation mechanisms.

SLAP, the *Secure Link Access Protocol*, is responsible for protecting all traffic between a client and its current access point. SLAP uses AES and HMAC-MD5 for encryption and sender authentication, respectively, and also provides a replay detection mechanism. The keys used by these services are established by a SIAP handshake and informed, via `ioctl`, to the SLAP module.

Compared to the solution proposed by 802.1X and WEP, our approach extends the services provided in many significant ways. First, the specification of the SIAP protocol permits interoperability between domains, mutual authentication between the mobile client and the access points in the network, and user-transparent mobility between networks with different IP prefixes. Second, by coalescing authentication and IP address assignment, SIAP enables the implementation of differentiated network views based on the IP address given to the client and also avoids the DoS attacks that can be performed against DHCP. The architecture has been shown also to avoid other DoS attacks present in 802.1X installations [18]. As the client's IP address becomes tied to its session key, the APs can identify and block any kind of address spoofing. Finally, having the client establish different session keys with each access point eliminates the need for an inter-AP protocol and prevents the propagation of state to access points not reachable to the client.

Besides the above design of architecture and protocols, our accomplishments under this contract include the implementation of both SIAP and SLAP in Linux 2.4 and the conducting of a number of experiments to show the viability of the architecture and the implementation. Even though a SIAP handshake involves Diffie-Hellman and RSA operations, we have used traces of existing wireless installations to show that the rate of SIAP transactions per unit of time handled by an access point can be sustained with very limited hardware, such as an outdated 333-MHz Pentium II processor. For this measurements we have used publicly available traces of three networks, from an over-provisioned network with 1366 users and 177 access points, to a popular conference installation with 195 users and 4 access points. These results are promising, giving the fact that clients can use very secure parameters, establishing

[5] With the unavailability of a global public-key infrastructure (PKI), mobile clients can use a different tuple for each network they have access to.

different session keys with every access points using 2048-bit RSA keys and 2048-bit primes for Diffie-Hellman operations.

SLAP per-frame overhead was measured to vary between $10\mu s$ and $330\mu s$ depending on frame size and for a hardware configuration using a 333-MHz Pentium II access point and a 900-MHz AMD client. The total overhead, which takes into account processing in both client and AP, was shown not to degrade the user's experience. We performed several 50-megabyte file transfers from servers with round-trip time values varying between 1ms and 40ms and measured the total download time to increase between 7% and 17%. We found these results encouraging given that they were achieved with software implementations of both AES and HMAC-MD5 running over a dedicated 100-Mbps FastEthernet link, used to emulate a high-bandwith wireless link. We expect the overhead to be negligible when provided with a hardware implementation.

This work has dealt with protecting the network from access by unauthorized users. However, another threat to wireless networks is a flooding attack in which the attacker uses up all the network bandwidth. This in itself is impossible to prevent. However, this problem raised our interest in using location information that can be extracted from a wireless network to at least locate the source of the attack. This information is also valuable in detecting that a single client is responsible for a very large number of authentication requests, another form of attack. The following subsection describes our preliminary work in this area, going beyond the scope of the original statement of work.

10.2 Self-Calibrating User Tracking

Various papers in the literature have already shown that mobile users using wireless cards (e.g. IEEE 802.11) give away information about their geographic position by transmitting packets over a wireless link. Systems such as RADAR and other commercial solutions use signal attenuation estimates to calculate the position of a mobile client given a set of access points placed in the environment. The available solutions, however, suffer from a major drawback: their reliance on human interaction during the calibration phase.

Before users can be located in a wireless network, the location system must learn about the characteristics of the environment, given that the received signal is heavily affected by construction materials, placement of rooms, walls and other obstacles, as well as the number, location and orientation of users. Current location systems gather this information during calibration phase, during which a technician walks around the target environment collecting multiple measurements of signal strength. This data is then processed to generate a signal map, which is then used by the location system to estimate the position of mobile clients. Published works differentiate themselves in how they process the collected data in order to generate what we called the signal map.

We advocate the construction of a user tracking system that calibrates itself without human assistance. The problem them becomes: what kind of accuracy can we achieve?

In our work, we investigate building a system that depends on a single piece of information: the location of the access points. Access points then should be able to acquire the propagation information they need by listening to each other and to mobile clients as they use the network[6].

One motivation for our approach is that access points are becoming increasingly inexpensive and overprovisioning a network with access points is a plausible idea[7]. It is easy to see that the complexity of the user location system decreases as the number of access points increases.

In our work up to the end of the contract, we developed a simulation of this self-calibrating location service and showed that, in the absence of large obstacles, the information about the placement of access points is enough to achieve good accuracy. We have for example simulated a 45x16m environment with 10 access points in which the system was able to locate clients with accuracy in the range of 1-3m most of the time. We are conducting experiments

[6] Note that information about mobile clients is approximate, as there is no calibration phase to provide signal readings for known locations in the environment.

[7] This idea is especially viable given we do not need complete access points to have a user tracking system. A simple device with a wireless interface to detect transmissions and a second interface to report the readings is sufficient.

to see the accuracy of the system in real installations and to define an access point placement methodology that guarantees a desired level of location accuracy.

10.3 Location-based Access Control

Assuming the location system described in Section 10.2, we then investigated how to geographically limit the access to a protected network. An accurate location system enables a network administrator to define geographically what is to be *inside* and *outside* its network infrastructure. In order words, wireless networks can be physically secured in almost the same way as the wired infrastructure. For example, with so-called war-driving attacks in which unauthorized users access wireless networks from a company's parking lot, the location system makes the attack far more challenging to accomplish because it now has to bypass the location system.

As mentioned ealier, the location system also provides a mechanism for detecting DoS attacks against authentication mechanisms. For example, consider again the architecture described in section 10.1. How does the network infrastructure know if a SIAP transaction request comes from inside or outside the company? A location-unaware system is clearly vulnerable to DoS attacks that generate a large rate of such requests. The location system enables the authentication server to control the rate of requests, dropping the messages that do not appear to come from inside the physically protected infrastructure.

There are numerous challenges related to integrating a location system with the authentication infrastructure. For example, can attackers fool the location system by the use of directional antennas and amplifiers? What is the rate of false positives? For example, a user could be physically inside but considered to be outside by the location system. How does this interaction affect the rate of requests satisfied by an authentication server? In on-going research and experimentation with real testbeds, we hope to show that these attacks can be prevented while keeping the false positive rate to a minimum.

In sum, this portion of the effort under this contract has developed technology for securely connecting to a network using name-based identification and providing mutual authentication. Relating back to TRIAD in general, as we expected, the address assignment to a client host is an insignificant portion of the overhead of authenticating the host, the latter being a necessary step. Therefore this address assignment, which obviates the need for an IPv6-based approach, is a trivial part of the authentication which is necessary in any case.

11 Virtual Private Networks

As a portion of the statement of work, we had proposed to investigate what was tentatively called WRAPSEC, a secure version of WRAP. We explored this portion of the effort in the context of implementing virtual private networks (VPN), including the additional aspect that the name/directory services should act as part of this private network as well.

As background, we recognize that the ability to channel packets through a secure tunnel between two endpoints in a standardized fashion is a necessity in a modern internet. Although a wide variety of different techniques exist to accomplish this, IPSec (ie, ESP in tunnel mode) is becoming the standard of choice. However, IPSec fails in the presence of NAT boxes because it attempts to secure communication from one IP address-identified endpoint to another. This is exposed within the IPSec standard as the interior IP packet is specified to have the "true" source and destination addresses, whereas the exterior IP packet has the addresses of the IPSec peers.

In our effort, we showed that one can use WRAP-in-WRAP in conjunction with the IPSec methodology to enable IPSec to traverse multiple realms. From the interior of the IPSec wrapper, the path appears to traverse a relay agent which corresponds to the pair of IPSec peers. To the outside world, the path appears to originate at one IPSec peer and terminate at the other. The IPSec peers (the nominal endpoints of the packet during its tenure in the untrusted portion of the internet) can further classify the packet based on the SPI to identify to which tunnel a given packet belongs, and thereby what set of keying and forwarding information to use.

Our approach has several appealing advantages. By multiplexing on the SPI of the IPSec tunnel, we allow an IPSec peer to use a single external address to handle multiple tunnels. This is a useful property because external addresses are typically assigned by an upstream provider, and gaining additional addresses may be a costly endeavor. Another useful property of our solution is that we have not exposed any additional information about a given flow, and therefore the various TFC (traffic flow confidentiality) tricks which IPSec allows may still be used. Finally, the IPSec tunneling is transparent to all hosts involved except the IPSec peers. This means that end hosts need not realize that their packets are being sent over a secure tunnel, since the tunnel ultimately appears to the end user as a relay agent.

A key accomplishment in this aspect of the work is the design of this approach and a reference implementation of this protocol using AES and HMAC-SHA1 as a Linux kernel module. However, further mechanisms is required to allow a client to establish or discover such a tunnel and then learn the associated addresses. Additionally, for better security we would like to periodically renegotiate the tunnels so that they will be rekeyed. However, we would also like to avoid heavy mechanism for tunnel negotiation - again, ideally we would like for the IPSec tunnel to be as transparent as possible for the end hosts.

To this end, we further modified the TRIAD NBR name/routing system to include tunnel discovery and establishment. To briefly illustrate its operation, assume for the moment that a secure tunnel exists between each pair of relays interested in serving as IPSec peers. Then, it is sufficient to simply relay and publish name lookup requests in the manner suggested by TRIAD, with the tunnel being treated as a standard peering relationship with another nameserver. The nameservers must perform some level of access control, of course, to prevent the names published by their secure tunnel peers from being exposed to the outside world, but the semantics we would like for these access controls actually match what is needed anyway for traditional naming purposes. When a name request is serviced using a secure peer, the nameserver can choose whether to establish a new tunnel or reuse an existing tunnel to this peer (it is probably wise to preserve a separate tunnel just for nameserver-nameserver communication in any case).

To solve the problem of the initial establishment of these tunnels between nameservers, we observed that in fact, most tunnels of interest are explicitly planned in advance. For instance, in a traditional VPN scenario, an employee (who is known to the corporation) knows the identity of the corporate portal. Thus, it is reasonable to expect the employee to instruct her nameserver to peer with the corporate portal. Additionally, it is reasonable to believe that the employee can unambiguously determine the public key of the corporate portal, and vice versa. Similarly, for a provider-provisioned VPN, as might exist between two branches of a corporation, there is inevitably some umbrella organization (in the case of branches of a single organization, the headquarters for the corporation; in the case of some number of collaborating organizations, the entity which manages the collaboration) which can, and should, deal with key distribution.

Furthermore, we argue that because DNS names are inherently hierarchical, tunnels between peer nameservers will only need to be established when they are actually being used. For instance, when *bob.tulsa.biz.com* tries to connect to *jane.omaha.biz.com*, the resolution of *omaha.biz.com* can cause the tunnel to be set up between *tulsa.biz.com* and *omaha.biz.com* without needing to also attempt to connect to the various other branch offices of *biz.com*. We also use the implicit DNS hierarchy to refer some subset of the *biz.com* namespace through the central headquarters, rather than establishing direct tunnels, giving us a fine-grained control over how the tunneling scheme is actually used. Finally, DNS aliases allow us to do things like cause *www.biz.com* to resolve as *www.omaha.biz.com*, thus hiding (to some degree) the exact location of network resources without actually affecting the tunneling needed to reach them.

In the work under this contract, we designed the above VPN mechanism and have a preliminary implementation with further refinement and extension taking place as part of on-going research. In this implementation, nameserver-to-nameserver tunnels use public keys for authentication plus the WRAP/IPSec implementation for actual communications. The implementation also exploits some of the cryptographic results from SLAP/SIAP work described in Section 10.

In summary, we expanded the original statement of work to address security at not just the network layer but also at the naming/directory layer. We have solved the single-realm limitations present in IPSec by nesting the

WRAP headers which allow us to solve similar limitations in IPv4. Additionally, we have developed a system for practical establishment and use of this tunneling functionality, for such applications as both traditional and provider-provisioned VPN. By leveraging the power of naming, we are able to avoid much of the heavyweight mechanism requisite in other solutions to these problems, without sacrificing flexibility or performance.

12 Named-Based Source-Controlled Multicast

Multicast is an important part of networking capability and may well be even more important in the future for efficient multi-point delivery. IP multicast is very much based on IP addresses and suffers from a limited multicast address range, particularly with the current standard "group" model of IP multicast. In particular, there is only 28 bits of addressing or 256 million groups world-wide for all application instances. Moreover, multicast is not designed to traverse NAT boxes.

As part of our work under this contract, we explored name-based multicast [19] to solve the above concerns, focusing on the more recent single-source multicast which is now being standardized and which originated from this research group under an earlier effort. As part of this investigation, we further realized that even this more recent standard suffers from the aspect of the layer 3 subscription mechanism being under the control of the receiver whereas applications really want this control to originate from the source, especially given the most common deployment scenarios which have the source as paying for the multicast delivery.

12.1 Name-Based Multicast

With name-based multicast (NBM), a multicast channel (to use SSM terminology) is identified by name, rather than by addressing, in contrast to the (S,G) normally used at this level. This naming can be realized as an extension of DNS or a host-level naming scheme because, with SSM, the identification of the multicast channel can be thought of properly as relative to an end host, namely the source. That is, an SSM channel is really identified as the G channel originating from host S. Thus, for example, a URL-like construct such as

```
http://video.darpa.mil//ipto/pimeeting03
```

could identify a multicast channel of an IPTO PI meeting at DARPA.

As part of the work on this contract, we developed a design for NBM, implemented a prototype of this design and demonstrated NBM operating through NAT realms. Following the general theme in this contract, this work shows that multicast can be extended independent of network-layer address translation. It also demonstrates how multicast can be provided as part of the directory system, solving a separate problem, namely how to identify available multicast channels in a user-friendly form.

12.2 Source-Controlled Multicast

We identified several problems with the current receiver-controlled multicast joining model.

First, there is a mismatch between multicast application designs/requirements and the receiver-oriented L3 multicast service model, resulting in security holes, inefficiencies, and a lack of information at the channel source. Multicast applications and higher-layer protocols, especially in the commercial realm, want the sources to have both information about receivers and control over the multicast session.

PGM poll messages, the EXPRESS counting mechanism, and the IGMPv3/PIM-SM Member Counting Proposal (NGC 2002) are all examples of sources attempting to gather information about receivers, in conflict with the receiver anonymity present in the L3 SSM model. As another example, TCP-SMO occasionally wants to eject a receiver from a multicast channel, however there is no mechanism allowing it to have this level of control. The most that it can do is to politely ask the receiver to leave the channel; it has no way to enforce this request.

A sensible multicast billing model involves the channel source paying its ISP for multicast, probably on a per-receiver basis, and in turn charging receivers for content. This is because the content provider is the one that benefits the most from multicast. Given this economic model, the source should be able to prevent non-paying receivers (detected at app layer) from being counted as receivers at the network layer. Otherwise, the source is left open to what is essentially a denial of service attack.

The existing multicast architecture involves numerous protocols (IGMPv2, IGMPv3, MSNIP, PIM-SM, MSDP, etc.) and much associated complexity, yet the world is converging to a web-based model for virtually all Internet communication applications. There has been an evolution from full anonymity between multicast senders and receivers in ASM to receiver anonymity only in SSM. However, even this partial anonymity is damaging to the deployment of secure commercial multicast applications, and is one of the main reasons for this complexity.

The Internet is divided into numerous discrete islands of multicast connectivity, thus application writers cannot rely on there being a global multicast service allowing an arbitrary source to reach an arbitrary receiver.

To address these problems, we developed a new source-controlled SSM IP option. It addresses the problems described above by:

1. Explicitly assuming a web-like model in which there is mutal knowledge and communciation between multicast sources and receivers prior to L3 multicast subscription and then taking advantage of this assumption by piggybacking multicast forwarding state establishment control messages on this higher level (http) communication for reduced latency.

2. Allowing the source to establish multicast forwarding state instead of receivers. This puts control in the hands of the entity that benefits the most from (and hence likely pays for) multicast service, prevents receivers from subscribing to nonexistent and inactive channels, and allows the source to reject any attempted subscription by an unauthorized receiver.

3. Supporting dynamic tunneling across multiple multicast clouds to provide a global and incrementally deployable SSM infrastructure.

As part of on-going research in this area, we are exploring several conjectures about SC-SSM, namely:

1. DoS Attack Resistance: We hope to show that, given an intelligent subscriber quota policy, SC-SSM is resilient in the face of DoS attack.

   The intuition behind this expected result is that SSM allows receivers to subscribe to nonexistent channels and doesn't allow sources to reject subscriptions from known bad or nonpaying receivers, while SC-SSM only allows subscriptions to active channels and allows a source to reject any attempted subscription.

2. Lower latency: Assuming a web-oriented model of the form in which a receiver accesses a multicast channel in the following steps:

   (a) Receiver performs name resolution on source's web server.

   (b) Receiver interacts with source's web server and, if successful, is provided with channel address (S, G).

   (c) Receiver issues IGMPv3 subscription for (S, G).

   then we see that regular SSM requires 3 round trips (1 for name resolution, one for http request and response, 0.5 for IGMPv3 join, and 0.5 for multicast data to arrive after join). However, we can easily show that the SC-SSM option, by virtue of piggybacking multicast forwarding state establishment on the http response which can be immediately followed by multicast data, eliminates one round trip for a latency savings of roughly 33% (minus processing time at the name server and the source's web server).

3. Scalability and Fast Crash Recovery: We expect to show that, despite per-receiver state at the channel source and periodic control messages to support fast crash recovery, SC-SSM scales to a large number of receivers. Quantitatively, we fix an acceptable (small) crash recovery time and (small) fraction of bandwidth that can be used for control traffic and show that we can support N receivers without exceeding these constraints. If we increase the allowable crash recovery time, then we reduce the number of periodic control messages and thus the percent of total bandwidth devoted to control traffic, allowing the source to support more receivers.

Besides these properties, we are see SC-SSM as providing the following benefits:

1. Full Receiver Information Available to Source: Under the regular SSM model, the channel source knows nothing about the receivers. With the current MSNIP proposal, the source learns 1 bit of information (whether or not there are any receivers). There are L3 proposals and a mechanism in the PGM reliable multicast protocol allowing the collection of an aggregate receiver count. It is clear that sources desire this info and more. SC-SSM goes beyond all of these proposals and allows collection of full receiver info, comparable to TCP-SMO or multiple unicast.

2. Dynamic Tunneling Mechanism Bridges Multicast Islands: The dynamic tunneling mechanism supported by SC-SSM allows multiple multicast islands to be connected, providing global multicast connectivity without the need for manual configuration of individual tunnels. Unlike other tunneling proposals, SC-SSM supports more than one tunnel on the path from source to receiver and doesn't require dependence on third party entities (e.g., tunnel servers).

3. Simplicity: SC-SSM requires fewer protocols, fewer lines of code and thus less to test and fewer chances for things to go wrong. It is easier to train humans to manage, so has less overall cost.

Overall, SC-SSM integrates well with NBM and the TRIAD architecture by relying on naming at the source and vesting control with the source. Moreover, it works well with the TCP extension described earlier supporting multicast because with this TCP extension, it is the source that controls what is multicast to whom and of course the connections are fundamentally identified by the names of the end points.

13 Naming in Future Internet

As another aspect of the work, the PI explored more generally the role of naming in the future Internet by looking at the overall requirements for name/directory support, the direct implementation implications, the costs and the viability of any other form of end-to-end identification such as end-to-end addresses or UUIDs.

The conclusions, as described in our report [20], are that the requirements on naming going forward effectively force the directory system to be coupled with the routing infrastructure for availability and performance, along the lines we have explored. Moreover, the costs of maintaining such as global directory system are such that any approach advocating a second such identification system as well would significantly increase the cost of the system, where cost considers overall complexity and management, not just equipment.

Furthermore, the IP address is shown to have lost any sense of semantics as part of becoming dynamically assigned, as are most IP addresses with the wide deployment of DHCP. The basic problem is that there is no assurance of how long an address really corresponds to the DNS name typically used to look up this address. With DHCP, this can change at any time. Thus, the name-based checksum is not just a feasible approach for dealing with NAT. It is in fact necessary to provide end-to-end reliability in the current Internet.

Finally, this work demonstrates that it is infeasible to provide a secure name service based on conventional "secrecy"-based approaches, i.e. keys, because the security then fails based on compromise of this secrecy, an unacceptable situation. Stated another way, a secure name service would require it to be trusted world-wide as an entity, which seems infeasible. Instead, we advocate making the name service highly available, treating any attempt at an attack or compromise as a denial of service effort, and relying on end-to-end authentication to detect when the name service has been compromised or failed, directing the connection to the wrong endpoint.

While this investigation was not directly identified in our statement of work, it is clearly part of developing the understanding of the Internet architecture directions and its conclusions support the name-based approach we have taken in TRIAD.

14 Technology Transitions

We have presented our work on Secure Internet access to Cisco and engendered some interest in this direction. We have distributed source code and reports to other researchers. To date, a talk on TRIAD has been presented at Cisco, Microsoft, Stanford, Bob Braden's End-to-end Research group, Symposium on Operating System Principles and at Active Networks and NGI PI meetings.

Talks on feedback-based routing were given at the International Computer Science Institute (ICSI), HotNets-I, Stanford Netseminar and AT&T Research Lab.

There has been on-going collaboration with Cisco to produce a Linux version of IGMPv3 to further the deployment of SSM using IGMPv3 and a subset of PIM-SM. A former member of the research group is now working full-time for Cisco and is involved in further IETF work on SSM.

15 Funding, Equipment and Personnel

We summarize aspects of the funding, equipment and personnel over the life of the project.

15.1 Funding

Rather than rely exclusively on (D)ARPA resources for all our support, we actively sought and received additional sources of funding from industry that provided greater leverage on the funding. Several sources of funding have provided for more equipment and staff than provided for in the contract. IBM, Intel and Microsoft have donated equipment and software. I believe that this funding has amplified the results we have achieved under the DARPA funding and provided funding for support personnel, allowing us to make our software more complete, more reliable and more efficient, i.e. faster. We also feel that it is speeding the transfer of the technology to industry where it can then become available in products to meet military requirements.

15.2 Equipment

Over the course of the contract, we have purchased and acquired through donations a variety of workstations, high-end PCs and server machines plus a switched hub and firewall. The equipment budget has been stretched as far as possible to using discounts, manufacturer equipment grants, etc. to provide a research environment of maximal effectiveness for the type of research we have been performing and plan to continue in the future.

15.3 Personnel

The contract has supported the PI, 10 Ph.D. students and several masters students in various degrees of support. Of these Ph.D. students, most have either completed their theses, or published significant research reports. We have been very successful in attracting good students as research assistants to the project. The number of students involved has been greater than the funding level would support because a number of the students we were initially supporting have now been awarded fellowship support (Stanford Graduate, Hertz, IBM, NSF fellowships). Fortunately, these students have continued to contribute to the goals of the project even after being awarded fellowships.

In summary, a major contribution of the contract has been to transfer the innovative ideas and techniques of this work through a large number of young, highly capable students and engineers to industry.

15.4 Publications

The publications generated during the project are included here as the references for the above summaries of our work.

References

[1] D. R. Cheriton and M. Gritter, TRIAD: A New Next-Generation Internet Architecture, http://www.dsg.stanford.edu/triad/triad.ps.gz, July, 2000

[2] D. Cheriton, TCP Name-based Checksum Option, Internet-draft, Distributed Systems Group, Stanford University, 2000.

[3] J. Stone and C. Partridge, When the checksum and the data disagree, SigComm 2000, August 2000.

[4] J. Stone, Performance of Checksum in the Internet, Stanford Computer Science Ph.D thesis, 2001.

[5] S Liang and D. Cheriton, Exploiting Fast Retransmit with Real-time Streaming Protocols, submitted for publication 2003.

[6] B. Yang, Phone Server: Design, Implementation and Performance Evaluation, Distributed Systems Group report, Sept. 2001.

[7] S. Liang and D. Cheriton, TCP-SMO: Extending TCP to support Medium-scale Multicast Applications, InfoComm, 2002.

[8] Unifying the Transport Layer of a Datagram Internetwork, S. Liang, Computer Science Ph.D. Thesis, Stanford University, 2003

[9] D. Cheriton and C. Rai, Wide Area Relay Addressing Protocol (WRAP), draft specification, March 2000.

[10] K. Argyraki and D. Cheriton, Active Internet Traffic Filtering: Real-time Response to Denial-of-Service Attacks, submitted for publication, 2003.

[11] M. Gritter and D. R. Cheriton, An Architecture for Content Routing Support in the Internet, Proceedings of USITS, 2001, March

[12] M. Gritter, Name-Based Routing, Stanford Computer Science Ph.D. Thesis, in preparation.

[13] M. Gritter and D. Cheriton, Denial-of-Service Attacks Against Internet Naming, in preparation, 2003.

[14] D. Zhu and D. Cheriton, Feedback-based Routing, WorkinProgress session, 3rd Usenix Symposium on Internet Technologies and Systems (USITS0́1), March 2001.

[15] D. Zhu and D. Cheriton, Feedback-based Routing, First Workshop on Hot Topics in Networks (HotNets-I), November 2002. (slides available on-line at http://www.stanford.edu/d̃apengz/.)

[16] D. Zhu and D. Cheriton, Feedback-based Routing, submitted for publication, 2003.

[17] D. B. Faria and D. R. Cheriton. Public-Key-Based Secure Internet Access (extended abstract), In *ACM Mobile Computing and Communications Review (MC2R)*, to appear. Selected posters from Mobicom'02.

[18] D. B. Faria and D. R. Cheriton. DoS and Authentication in Wireless Public Access Networks, In *Proceedings of the First ACM Workshop on Wireless Security (WiSe'02)*, pages 47-56, Sept. 2002.

[19] V. Laviano and D. Cheriton, Name-based Multicast, Proc. NGC, 2003.

[20] D. Cheriton, Scalability and Naming in the Internet Architecture, submitted for publication, 2003.

15.5 Other Activities

The PI served as a consultant for a number of organizations interested in network topology general networking issues, including Cisco Systems, Google, Sun and Zambeel.

16 Follow-on Activities

Through this work, we have realized the major driving consideration in a revised Internet architecture has to be security and attack resistance. This is evident in our initial work on secure Internet access, feedback-based routing and filter propagation.

17 Concluding Remarks

The TRIAD project was a highly ambitious research effort to investigate the real problems with the Internet architecture and develop solutions for the next generation. It would have been tempting to develop a radical new departure from both IPv4 as well as IPv6, and glibly call for throwing out the current Internet protocols. As much as we have explored a full rethinking of the overall architecture, we have been conscious of the value of maintaining as much of the existing Internet architecture as feasible. In particular, if an "optimal" redesign of the network layer datagram format provides a 5 to 10 percent improvement over IPv4, it is simply not worth changing. We want to focus our energies on the areas in which there is true value in changes.

The consequence of this exciting venture was a large and diverse collection of research results, protocols and hardware/software realizations that have wide applicability to US military applications and to the commercial arena. We have summarized these results in this report and referenced the appropriate documents. Moreover, we believe we have gone well beyond the contract deliverable in exploring aspects of attack-resistance, feedback-based routing and virtual private networks.

Of equal importance to our research resuls and deliverables, we have produced a significant number of Ph.D. students in experimental Computer Science, addressing a pressing national need. In fact, lack of technical people in Computer Systems is recognized as a significant issue in addressing next generation military projects. The quality of education of these students and the quality of the research in general would have been impossible without the funding support and careful research management we have enjoyed from DARPA on this project.

We have also invested more than typical in good software and hardware development, but produced more than normal return in the form of: (1) utility of the prototypes, (2) depth of experience with our ideas and their implementation, and, (3) transfer of technology to others. As a consequence, software implementing the protocols described in this report and our publication has been distributed to various universities, military sites and companies.

In addition to the immediate results of the contract work, we have created an research group and environment that is capable of (and of course willing to) take on advanced research in further areas of communication, parallelism and distributed computation of interest to DARPA and required for future military applications.

18 Acknowledgements

I wish to thank David Tennenhouse for taking the original initiative to fund this work, and to Mari Maeda who took over as program manager and guided and supported us further to make a significant contribution to DARPA's goals in the areas of networking and distributed computing.